UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Efficient motion planning for high DOF hands using principal motion directions

J. Rosell, R. Suárez, C. Rosales, J. A. García and A. Pérez.

IOC – Divisió de Robòtica

IOC-DT-P-2008-13
Octubre 2008

Institut d'Organització i Control
de Sistemes Industrials

Efficient motion planning for high DOF hands using principal motion directions

J. Rosell, R. Suárez, C. Rosales, J. A. García and A. Pérez

Abstract

The paper deals with the problem of motion planning of anthropomorphical mechanical hands avoiding collisions. The proposed approach tries to mimic the real human hand workspace, but reducing the dimension of the search space in order to obtain results as a compromise between motion optimality and planning complexity (time) by means of the concept of principal motion directions. Basically, the work includes the following phases: capturing the human hand workspace using a sensorized glove and mapping it to the mechanical hand workspace, reducing the space dimension by looking for the most relevant principal motion directions, and planning the hand movements using a sampling-based roadmap planner. The approach has been implemented for a four finger anthropomorphical mechanical hand, and some examples are included to illustrate its validity.

I. INTRODUCTION

Advances in robotics are producing a number of complex devices with a high number of degrees of freedom (DOF), lots of sensors, and sophisticated controllers to assure stability and a good performance. These devices include different types of robots, adapted to different environments and tasks, and among them the most representative instances are the humanoids, equipped with anthropomorphic hands with a number of DOF ranging from 12 (four fingers with 3 independent DOF each one) to 25 (some models consider 4 DOF in each finger and several DOF in the palm [1]). Examples of anthropomorphic hands with four fingers are the Utah/MIT Hand [2], DIST Hand [3], LMS Hand [4]; DLR Hand [5] and MA-I Hand [6], and examples with five fingers are the Belgrade/USC Hand [7], Anthrobot-2 Hand [8], NTU Hand [9], ROBONAUT [10], Shadow Hand [11], Gifu Hand [12] and Bolonia Hand 3 [13]. Good discussions about robot hands can be found in [14] and [15].

Despite the advanced features of these mechanical hands, one of the remaining problems in order to obtain a good outcome from them is the automatic determination of their movements, which are quite complex and non evident for the human being in the space of generalized coordinates. This is a well known motion planning problem, but in a really large dimensional space, thus some new approaches are still necessary in order to find solutions that can be really implemented and used in practice. This paper presents some developments in this line, looking for procedures that allow the automatic motion planning of anthropomorphic hands in a smooth way, caring for collisions with the environment as well as between the different parts of the hand.

II. PROBLEM STATEMENT AND SOLUTION OVERVIEW

The basic problem to be solved is the following: given the current hand configuration C_o and a final desired one C_d (that is not necessarily a grasping configuration), find a collision free path from C_o to C_d. The dimension of the configuration space of this problem is equal to the number of DOF of the hand, thus it is hard to solve it in a conventional way without a high computational time. In this context, the proposed approach is based on a reduction of the configuration space dimension, which is done by looking for a representative subspace $\mathcal{SC}$ of the whole hand configuration space $\mathcal{C}$, and looking for continuous valid paths in $\mathcal{SC}$. Of course, there may be solutions in $\mathcal{C}$ not included in $\mathcal{SC}$, thus the selection of a proper subspace $\mathcal{SC}$ is a relevant step in the proposed approach. On the other hand, if a solution is found in $\mathcal{SC}$, for sure it is valid in $\mathcal{C}$.

The main consideration that supports the reduction of the problem space is that the hand has several joint movements that are not (completely) independent, and therefore they can be associated in some way. A typical example is given by the last two joints of each finger in the human hand, which (normally) cannot be moved independently; in the same way some other relations can be found analyzing the hand configuration space. This analysis is done by taking enough samples of the hand configuration space $\mathcal{C}$ and looking then for the direction in which the samples present the largest dispersion, which is iteratively repeated considering orthogonal directions until a new basis of $\mathcal{C}$ is generated. Then, by selecting the n first found vectors of this basis and properly choosing

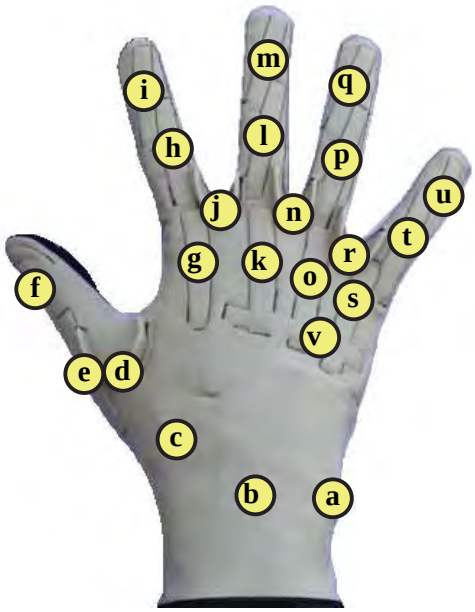


Fig. 1. Sensorized glove used to capture the operator hand workspace.

a bounding box aligned with these vectors and centered in the mean value of the original set of points, a good bounded approximation of $\mathcal{C}$ is found.

A relevant previous work in this line [16] uses an initial set of grasping configurations to find a bidimensional *grasp subspace*, i.e. to characterize the configurations of the hand used to grasp different objects. As a difference with that work, we use here an initial set of unconstrained general hand configurations in order to model all the real hand workspace and not only potential grasping configurations. The particular procedure followed to generate the set of hand configurations as well as to find a proper bounded subspace is detailed in Section III. Once the bounded subspace is determined, sampling-based motion planning techniques are used to model the free configurations and to find free paths between any two of them (Section IV). Other works using the concepts introduced in [16] uses a bidimensional subspace to look for grasping configurations [17], [18].

The approach followed in this work can be summarized in the following steps:

1) Use of a sensorized glove to obtain samples of the human hand workspace (22 DOF) (Subsection III-A).
2) Map these samples to the configuration space of a mechanical hand $\mathcal{C}$ (13 DOF) (Subsection III-B).
3) Find a representative subspace $\mathcal{SC}$ of the mechanical hand configuration space $\mathcal{C}$ (between 3 and 6 DOF) (Subsection III-C.
4) Use a sampling-based roadmap planner to model the free space of the representative subspace obtained in the previous step (Section IV).
5) Finally, given an initial and final hand configurations C_o and C_d, connect them to the roadmap and use it to find a free path between them.

III. MODELING HAND MOVEMENTS WITH PRINCIPAL MOTION DIRECTIONS

A. Data Aqcuisition

The data acquisition is done using the commercial sensorized glove CyberGlove© from Immersion Corporation, shown in Fig. 1. It is a fully instrumented glove that provides up to 22 high-accuracy joint-angle measurements. It uses proprietary resistive bendsensing technology to transform hand and finger positions into real-time digital joint-angle data. The 22-sensor model has three flexion sensors per finger, four abduction sensors, a palm-arc sensor, and sensors to measure wrist flexion and abduction (Fig. 1).

Fig. 2. Human hand with the sensorized glove connected to the mechanical hand simulator used in the data acquisition procedure.

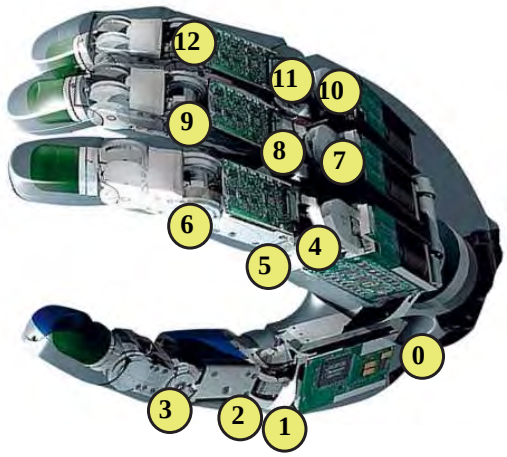


Fig. 3. The antropomorphic mechanical hand used.

After a calibration procedure for each user's hand, the movements captured with the glove are mapped to movements of the mechanical hand through a virtual simulator (Fig. 2), where the users can have a visual feedback of the mapping (detailed in the next subsection). Thus, the users move their hands in an unconstrained way covering the hand workspace. Each mapped posture is recorded for further processing and analysis.

B. Mapping from the Sensorized Glove to the Mechanical Hand

The Schunk antropomorphic hand (SAH), shown in Fig. 3, is a robotic hand with 13 DOF resembling a human hand in terms of mobility and appearance. It has four identical fingers and one is equipped with an additional joint to function as the opposing thumb. Each finger has four joints, although the distal joint is mechanically coupled to the middle joint, i.e. there are three DOF per finger.

Since the SAH mimics the human hand movements, mapping the data from the glove sensors to the movements of the SAH is done in an almost direct way. The following issues are considered for the mapping (see Figures 1 and 3):

- The palm of the mechanical hand is rigid and therefore the palm arc sensor v and the wrist flexion and abduction sensors b and a are ignored.
- The mechanical hand lacks the little finger and therefore the sensors u, t, s and r are ignored.
- The distal phalanx sensors i, m, and q are not used since the SAH hand has a coupling between the medium and distal phalanx of each finger.
- Using the sensor c for controlling joint 1 produces a more natural motion of the SAH than when using sensor d, therefore sensor d is ignored and sensor c is used for both joints 0 and 1.

TABLE I

CORRESPONDENCE BETWEEN THE CYBERGLOVE SENSORS (FIG. 1) AND THE JOINTS OF THE SAH HAND (FIG. 3).

Cyberglove Sensor		SA Hand Joint	
Id.	Name	Id.	Name
c	thumb roll	0	thumb base
c	thumb roll	1	finger base (thumb)
e	thumb inner	2	proximal phalanx (thumb)
f	thumb outer	3	medium phalanx (thumb)
j	index abduction	4	finger base (index)
g	index inner	5	proximal phalanx (index)
h	index middle	6	medium phalanx (index)
-	medium abduction	7	finger base (medium)
k	medium inner	8	proximal phalanx (medium)
l	medium medium	9	medium phalanx (medium)
n	ring abduction	10	finger base (ring)
o	ring inner	11	proximal phalanx (ring)
p	ring medium	12	medium phalanx (ring)

- The abduction is measured in a relative way, i.e. sensors j and n give the relative angle between the index-middle and middle-ring fingers, respectively. Then, the mapping is done using the middle finger as reference, i.e. the base of the middle finger (joint 7) is fixed to zero, which lets sensors j and n to be directly associated to joints 4 and 7, respectively.

Then, only eleven values from the twenty two available in the glove are used to control the joints of the SAH mechanical hand. The complete mapping is shown in Table I. Note that this mapping makes the motions of the SAH hand to be defined with 11 independent parameters, although it has 13 DOF.

C. *Principal Motion Directions*

Dimensionality reduction of a feature set is a common preprocessing step used for pattern recognition and classification applications and in compression schemes. Principal component analysis (PCA) is a vector space transform often used in these fields to reduce multidimensional data sets to lower dimensions for analysis [19]. It is also used as a tool in exploratory data analysis and for making predictive models. PCA involves the computation of the eigenvalue decomposition of a data covariance matrix or the singular value decomposition of a data matrix, usually after mean centering the data for each attribute.

In this work, PCA is used to reduce the configuration space of the mechanical hand SAH to a more tractable space of smaller dimension, using the data recorded from the hand postures. Fig. 4 shows some examples of the existing correlation between joints that illustrate that an effective reduction can be obtained.

The vectors that define the new base of the hand space are called Principal Motion Directions (PMDs). Selecting only the first vectors with higher associated variances a reduced hand space is obtained: The first PMD represents the 42.19% of the total variance in the dataset analyzed; the first two components the 77.12%, and the first three components the 84.71% (the complete evolution is shown in Fig. 5). Therefore, in this work the use of three PMDs has been considered sufficient. Fig. 6 shows the hand postures along the two principal components, and Fig. 7 the postures resulting from their combination.

In the Appendix the MatLab cose used for the PCA analysis is shown, as well as the coordinates of all the PMDs obtained and their corresponding variances.

IV. MOTION PLANNING

Sampling-based motion planners have demonstrated to be one of the best alternatives for path planning problems, since they avoid the explicit characterization of the obstacles of configuration space ($\mathcal{C}$). These planners generate collision-free samples of $\mathcal{C}$ and connect them with free paths capturing the connectivity of the free space either by forming roadmaps [20] or trees [21].

A sampling-based roadmap planner, using a deterministic sampling sequence as sampling source, will be used here to find the hand motions.

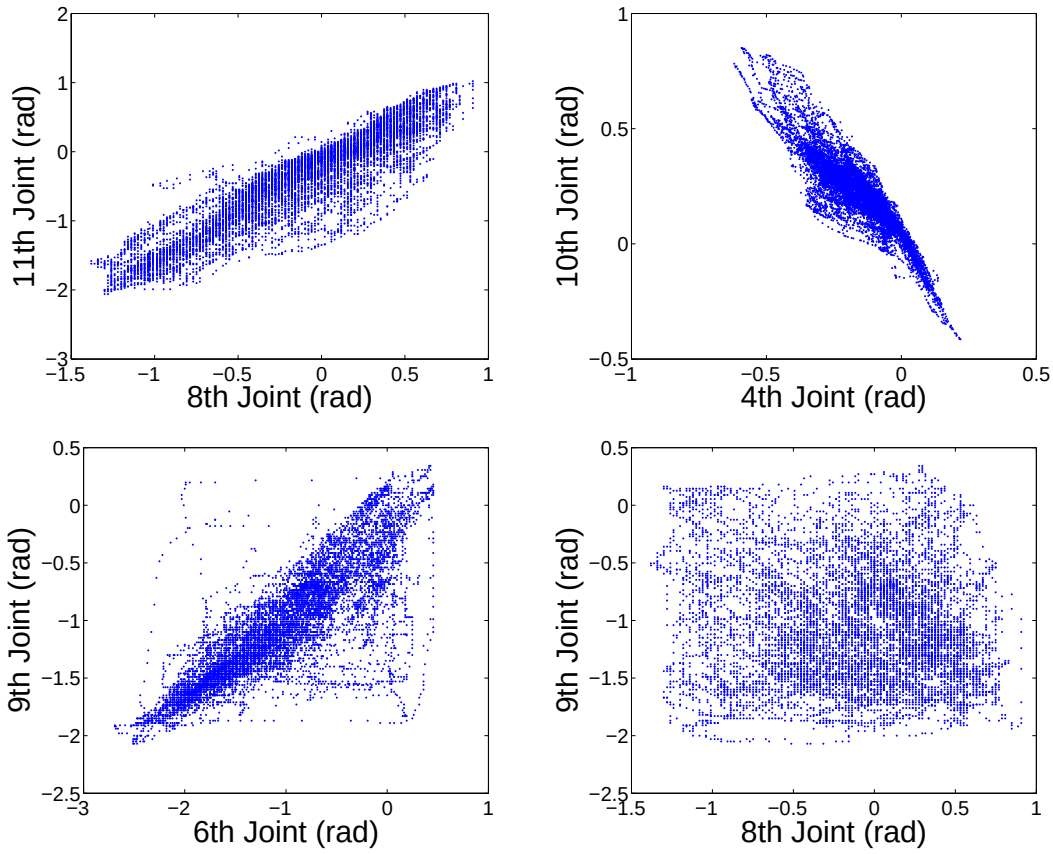


Fig. 4. Top-left: Positive correlation between proximal phalanxes (8 and 11); Top-right: Negative correlation between the abduction of the index and the ring (4 and 10); Bottom-left: Smooth positive correlation between medium phalanxes (6 and 9); Bottom-right: No correlation between consecutive phalanxes of the medium finger (8 and 9).

A. *The algorithm*

A basic sampling-based roadmap planner has an initial preprocessing phase to construct the graph that represents the roadmap capturing the connectivity of the free configuration space ($\mathcal{C}_{free}$). The nodes of the graph are the configurations sampled from $\mathcal{C}_{free}$ and the edges the collision-free paths that connect them.

Fig. 8 shows the algorithm that returns the graph G representing the roadmap. The input is the number N of configurations to sample. The functions used are:

- Function INSERT(s,V): inserts a node to the set V of vertices of a graph. The use of an efficient graph structure like the ones provided by the *Boost Graph Library* [22] greatly enhances the implementation.
- Function INSERT((s,q),E): inserts the edge (s,q) connecting configurations s and q to the set E of edges of a graph.
- Function GET-SAMPLE(): generates a configuration of $\mathcal{C}$; it is detailed in the following subsection.
- Function FREE(s): returns TRUE if the configuration s belongs to $\mathcal{C}_{free}$. The use of a simple and efficient collision detection library like *PQP* [23] is mandatory since this is a time-consuming step of the process.
- Function NEIGBORHOOD(s): returns the set of up to K samples that lie within a predefined neighborhood of s. This can be efficiently implemented using the MPNN algorithm [24] or taking advantage of the grid structure if grid-based deterministic sampling sequences are used [25].
- Function CONNECT(s,q): determines whether the rectilinear path connecting s and q is free or not by performing the collision-check test to several of its configurations. This can be efficiently done using the binary method [26].

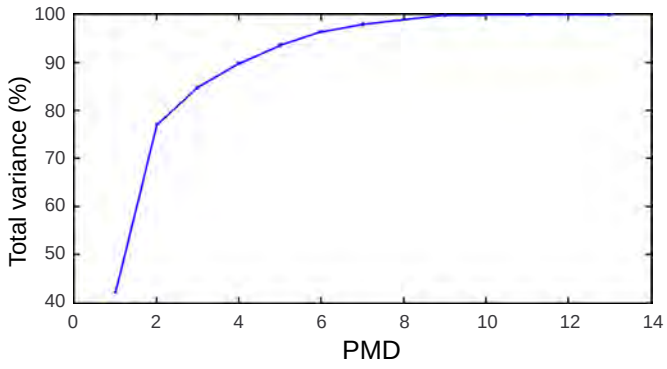


Fig. 5. Total variance covered when using an increasing number of PMDs.

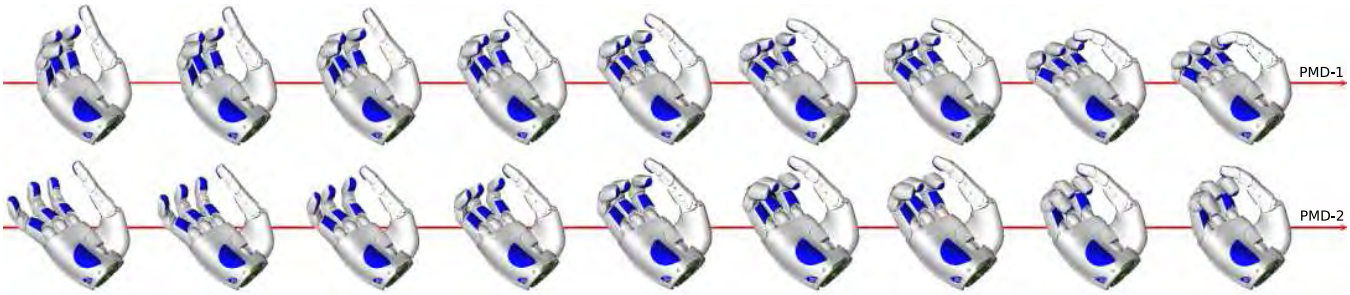


Fig. 6. Configurations of the SAH hand when moved along the first two PMDs.

B. The sampling source

Sampling-based methods usually rely on the use of a random number generation source, although the use of deterministic sampling sequences is a good alternative [27]. Deterministic sampling sequences provide an incremental and uniform coverage of $\mathcal{C}$, with a better dispersion than random sampling. Deterministic sampling has given slightly better results than random sampling in roadmap planners [28] (although Hsu et al. [29] constrained this improvement to few degrees of freedom tasks and considered it much less significant than the importance sampling issue devoted to bias the samples towards regions relevant to the task). With the objective of sampling uniformly over the space defined by the first main PMDs, the use of a deterministic sampling sequence is therefore a good option.

Let $\mathbf{e}_1, \ldots, \mathbf{e}_E$ be the first E PMDs when ordered in a decreasing order of their corresponding eigenvalues $\lambda_1, \ldots, \lambda_E$. The function GET-SAMPLE() uses the $s_d(k)$ deterministic sequence [30] to obtain a point of an E-dimensional unitary hypercube. $s_d(k)$ is based on a multi-grid cell decomposition with an efficient cell coding, as well as on the use of the digital construction method first proposed in [31]. The $s_d(k)$ sequence is a sequence of cells of the maximum partition level; the samples are random configurations within those cells.

Once an E-dimensional configuration $\mathbf{p} = (p_1, \ldots, p_E)$ is obtained with function GET-SAMPLE(), it is used as follows to compute the sample $\mathbf{\Theta} = (\theta_1, \ldots, \theta_d)$ of the d-dimensional joint space:

$$\theta_i = b_i + \sum_{j=1}^{E} (1 - 2p_j)\lambda_j \mathbf{e}_j \qquad i = 1 \ldots d \tag{1}$$

where b_i is the ith coordinate of the mean of the dataset (Section III-C).

V. EVALUATION AND DISCUSSION

Hands are usually mounted on robot arms, thus, for the experiments, a simple robot with three revolute joints has been added to the SAH hand. Is is shown in Fig. 9, where the initial and goal configurations (C_o and

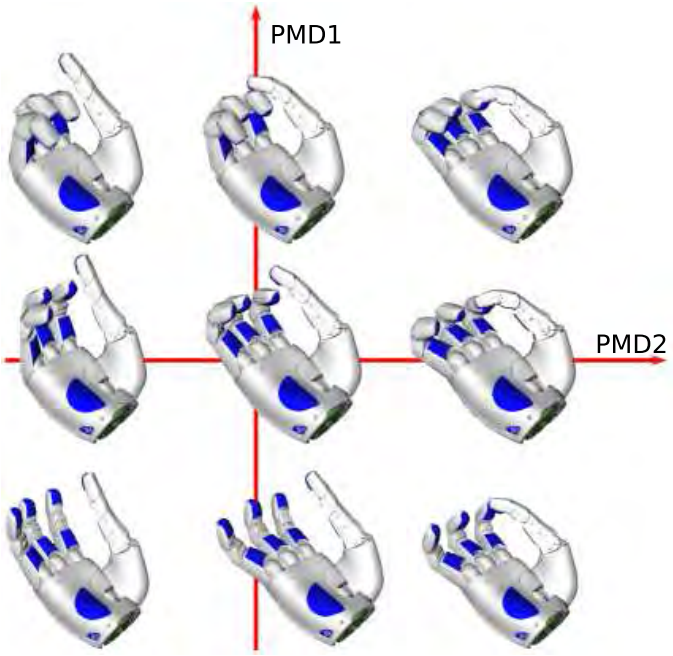


Fig. 7. Configurations of the SAH hand when moved along the combination of the first two PMDs.

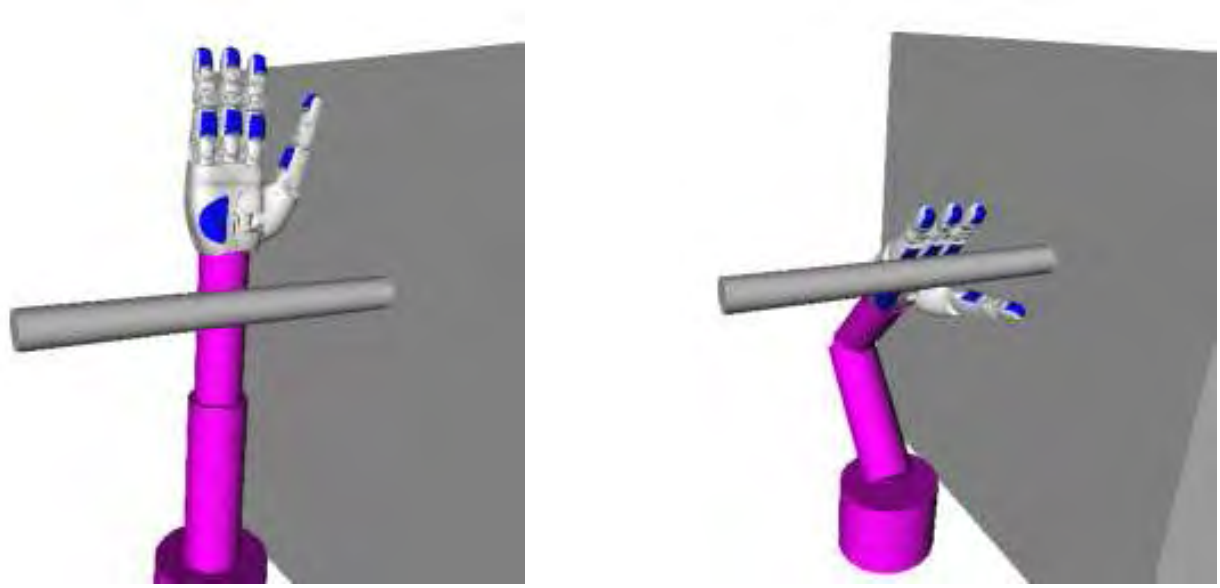
Basic Sampling-Based RoadMap(N)

$G.vertexSet \leftarrow \varnothing$, $G.edgeSet \leftarrow \varnothing$ $i \leftarrow 0$
For $i = 1$ to N do:
  $s =$ GET-SAMPLE()
  If FREE(s) then
    INSERT$(s, G.vertexSet)$
    ForAll $q \in G.vertexSet \mid s \neq q$ and $q \in$ NEIGBORHOOD(s) do
      If CONNECT(s, q) then
        INSERT$((s, q), G.edgeSet)$
      End If
    End For
  End If
End For
RETURN G

Fig. 8. Algorithm for the preprocessing phase of a basic sampling-based roadmap planner.

C_d) are illustrated. The proposed approach is validated using three PMDs by comparing its efficiency with a basic probabilistic roadmap planner sampling over all the joint space. Considering the 3 DOF robot, the reduced configuration space case, called 6-SDK, accounts for 6 dimensions, and the complete joint space case, called 16-RND, accounts for 16 dimensions. Using only 100 samples, the success ratio of the proposed approach is nearly three times larger than that of the basic probabilistic roadmap planner (see Table II). In order to obtain the same success ratio in both cases, the 16-RND case requires an average of 300 samples, making the computational cost due to the collision-check process nearly three times larger.

It is noticeable that reducing the dimensionality of the configuration space using PMDs allows to use sampling techniques in a more efficient way, i.e. enlarging the ratio of free samples over the total, and consequently reflected in the success. One good reason for this is that the sampling over PMDs provides free self-collision samples very often. Aside from the quantitative results, it is worth noting that the motions obtained with the proposed



a) b)

Fig. 9. Configurations: a) C_o and b) C_d.

TABLE II

COMPARISON BETWEEN APPROACHES FOR THE TEST CASE.

Variables	16-RND	6-SDK
Success (%)	18	55
# Samples	100	100
# Avg. roadmap nodes	36	89

approach look more natural from the anthropomorphical and aesthetic point of view, as it can be appreciated in Fig. 10 where two sequences of snapshots are shown, respectively, for the 16-RND and 6-SDK case. They are also been submitted as an accompanying video.

VI. CONCLUSIONS

This paper has presented an efficient methodology to compute the collision-free motions of a hand based on the modeling of the principal motions direction (PMDs). These directions, obtained by demonstration using pattern recognition techniques, capture the natural motion of the human hand. Taking the PMDs with more weight (those with a larger eigenvalue), allows a reduction on the dimension of the hand-movement space that greatly eases the work of a sampling-based roadmap planner. Added to a good computational efficiency, the hand motions obtained by the proposed planner are more natural than those obtained when all the degrees of freedom of the hand are directly sampled using a random sampling sequence in the joint space. Currently, the proposed planner is being enhanced by the use of an importance sampling method to bias the samples towards more relevant regions of the configuration space, in order to perform tasks with smaller clearances.

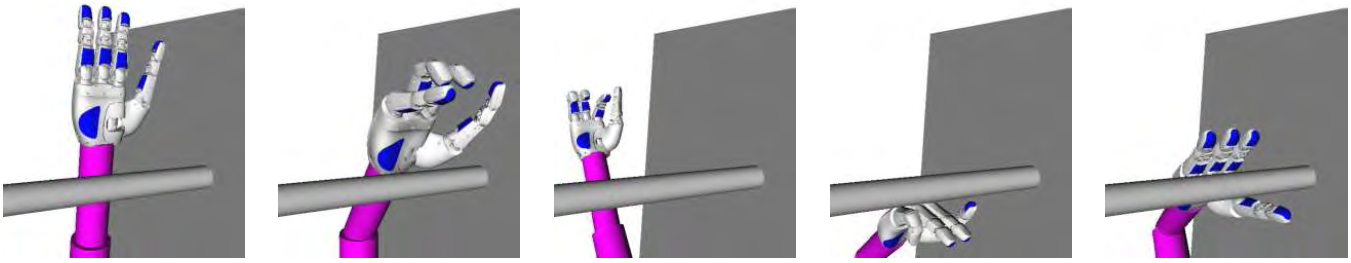
REFERENCES

[1] E. Peña, J. Yang, and K. Abdel-Malek, "Santos^{TM} hand: A 25-degree-of-freedom model," in *Proceedings of SAE Digital Human Modeling for Design and Engineering*, Iowa City, Iowa, USA, June 2005.

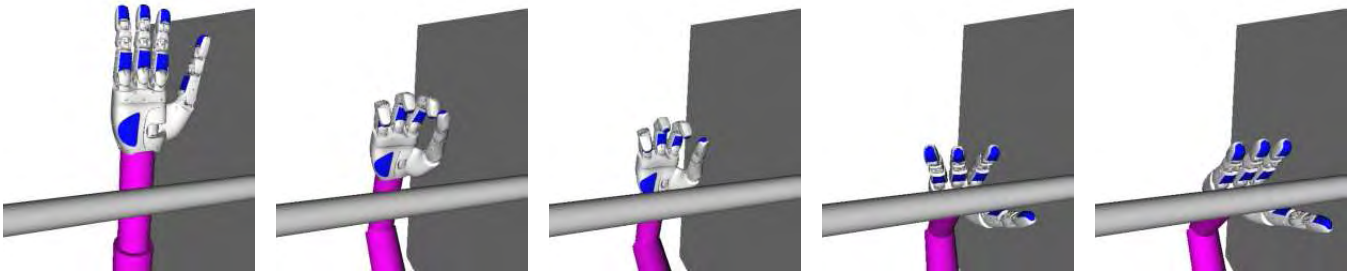
[2] S. C. Jacobsen, J. E. Wood, D. F. Knutti, and K. B. Biggers, "The UTAH/M.I.T. dextrous hand: Work in progress," *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 21–50, December 1984.

[3] A. Caffaz and G. Cannata, "The design and development of the DIST-Hand dextrous gripper," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, 1998, pp. 2075–2080 vol.3.

[4] J. P. Gazeau, S. Zehloul, M. Arsicault, and J. P. Lallemand, "The LMS hand: force and position controls in the aim of the fine manipulation of objects," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, 2001, pp. 2642–2648 vol.3.



a) Solution found with the 16-RND approach



b) Solution found with the 6-SDK approach

Fig. 10. Snapshots of one of the successful attempts to connect the given C_o and C_d configurations.

[5] J. Butterfass, M. Fischer, M. Grebenstein, S. Haidacher, and G. Hirzinger, "Design and experiences with DLR hand II," in *World Automation Congress, 2004. Proceedings*, vol. 15, 2004, pp. 105–110.

[6] R. Suarez and P. Grosch, "Mechanical hand MA-I as experimental system for grasping and manipulation," in *VideoProceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, April 2005.

[7] G. A. Bekey, R. Tomovic, and I. Zeljkovic, *Control architecture for the Belgrade/USC hand*. New York, NY, USA: Springer-Verlag New York, Inc., 1990, pp. 136–149.

[8] M. S. Ali, K. J. Kyriakopoulos, and H. E. Stephanou, "The kinematics of the anthrobot-2 dextrous hand," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, 1993, pp. 705–710 vol.3.

[9] L.-R. Lin and H.-P. Huang, "Mechanism design of a new multifingered robot hand," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, 1996, pp. 1471–1476 vol.2.

[10] C. S. Lovchik and M. A. Diftler, "The robonaut hand: a dexterous robot hand for space," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, 1999, pp. 907–912 vol.2.

[11] Shadow Robot Company, "Design of a dextrous hand for advanced clawar applications," in *Climbing and Walking Robots and the Supporting Technologies for Mobile Machines: CLAWAR 2003*, 2003, pp. 691–698.

[12] H. Kawasaki, T. Komatsu, and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 3, pp. 296–303, 2002.

[13] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, G. Palli, and C. Melchiorri, "Development of UB hand 3: Early results," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 4488–4493.

[14] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 6, pp. 652–662, 2000.

[15] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura, "How far is the human hand? a review on anthropomorphic robotic end-effectors," University of Bologna, Tech. Rep., 2004.

[16] M. F. M. Santello and J. F. Soechting, "Postural hand synergies for tool use," *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, December 1998.

[17] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 3270–3275.

[18] A. Tsoli and O. C. Jenkins, "2d subspaces for user-driven robot grasping," in *Proc. of the RSS 2007 Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

[19] J. I.T., *Principal Component Analysis*. Upper Saddle River, NJ, USA: Springer Series in Statistics, 2002.

[20] L. E. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1994, pp. 2138–2145.

[21] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 995–1001.

[22] L.-Q. Lee, J. G. Siek, and A. Lumsdaine, "The generic graph component library," in *14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications.*, New York, 1999, pp. 399–414.

[23] M. C. L. Eric Larsen, Stefan Gottschalk and D. Manocha, "Fast proximity queries with swept sphere volumes," in *Proc. of Int. Conf. on Robotics and Automation*, 2000, pp. 3719–3726.

[24] A. Yershova and S. LaValle, “Improving motion planning algorithms by efficient nearest-neighbor searching,” *IEEE Trans. on Robotics*, vol. 23(1), pp. 151 – 157, 2006.
[25] J. Rosell and A. Pérez, “Sampling C-obstacles border using a filtered deterministic sequence,” Technical report IOC-UPC-DT-P-11-2008, Tech. Rep., September 2008.
[26] M. H. O. Roland Geraerts, “Sampling and node adding in probabilistic roadmap planners,” *Robotics and Autonomous Systems*, vol. 54, p. 165173, 2006.
[27] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang, “Quasi-randomized path planning,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 1481–1487.
[28] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, “On the relationship between classical grid search and probabilistic roadmaps,” *Int. Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.
[29] D. Hsu, J.-C. Latombe, and H. Kurniawati, “On the probabilistic foundations of probabilistic roadmap planning,” *Int. Journal of Robotics Research*, vol. 25, no. 7, pp. 627 – 643, 2006.
[30] J. Rosell, M. Roa, A. Pérez, and F. García, “A general deterministic sequence for sampling d-dimensional configuration spaces,” *J. of Intelligent and Robotic Systems*, vol. 50, no. 4, pp. 361–374, 2007.
[31] S. R. Lindemann, A. Yershova, and S. M. LaValle, “Incremental grid sampling strategies in robotics,” in *Proc. of the Sixth Int. Workshop on the Algorithmic Foundations of Robotics*, 2004, pp. 297 – 312.

APPENDIX

The principal component analysis has been done using MatLab:

```
PRINCOMP Principal Components Analysis.
    COEFF = PRINCOMP(X) performs principal components analysis on the N-by-P
    data matrix X, and returns the principal component coefficients, also
    known as loadings.  Rows of X correspond to observations, columns to
    variables.  COEFF is a P-by-P matrix, each column containing coefficients
    for one principal component.  The columns are in order of decreasing
    component variance.

    PRINCOMP centers X by subtracting off column means, but does not
    rescale the columns of X.  To perform PCA with standardized variables,
    i.e., based on correlations, use PRINCOMP(ZSCORE(X)).  To perform PCA
    directly on a covariance or correlation matrix, use PCACOV.

    [COEFF, SCORE] = PRINCOMP(X) returns the principal component scores,
    i.e., the representation of X in the principal component space.  Rows
    of SCORE correspond to observations, columns to components.

    [COEFF, SCORE, LATENT] = PRINCOMP(X) returns the principal component
    variances, i.e., the eigenvalues of the covariance matrix of X, in
    LATENT.

    [COEFF, SCORE, LATENT, TSQUARED] = PRINCOMP(X) returns Hotelling's
    T-squared statistic for each observation in X.

    When N <= P, SCORE(:,N:P) and LATENT(N:P) are necessarily zero, and the
    columns of COEFF(:,N:P) define directions that are orthogonal to X.

    [...] = PRINCOMP(X,'econ') returns only the elements of LATENT that are
    not necessarily zero, i.e., when N <= P, only the first N-1, and the
    corresponding columns of COEFF and SCORE.  This can be significantly
    faster when P >> N.

    See also factoran, pcacov, pcares.


    Reference page in Help browser
       doc princomp
```

```
%Load sampling data, data 13.txt, in SpaceVar
SpaceVar=load('data13.txt');

%Creating some joint correlation charts
subplot(2,2,1),plot(SpaceVar(:,9),SpaceVar(:,12),'.')
xlabel('8th Joint (rad)','FontSize',20);
ylabel('11th Joint (rad)','FontSize',20);

subplot(2,2,2),plot(SpaceVar(:,5),SpaceVar(:,11),'.')
xlabel('4th Joint (rad)','FontSize',20);
ylabel('10th Joint (rad)','FontSize',20);

subplot(2,2,3),plot(SpaceVar(:,7),SpaceVar(:,10),'.')
xlabel('6th Joint (rad)','FontSize',20);
ylabel('9th Joint (rad)','FontSize',20);

subplot(2,2,4),plot(SpaceVar(:,9),SpaceVar(:,10),'.')
xlabel('8th Joint (rad)','FontSize',20);
ylabel('9th Joint (rad)','FontSize',20);


%Calculating the principal component coefficients, COEFF,
%the representation of SpaceVar in the principal component space, SCORES
%the eigenvalues of the covariance matrix of SpaceVar, LATENT.
[COEFF, SCORE, LATENT]=princomp(SpaceVar);
save PrinComp.txt COEFF -ascii;
save Scores.txt SCORE -ascii;
VARIANCE=LATENT.';
save Variances.txt VARIANCE -ascii;
```

The output of the principal contact analysis is:

```
−6.2567630e−002  4.6247381e−003 −7.1131013e−001 −1.4620041e−002  9.4565845e−002
−3.7937290e−001  5.4937742e−001  1.7866405e−001 −6.5758756e−003  2.0066585e−002
−1.0890991e−002 −5.2159812e−002 −0.0000000e+000

−3.8540126e−003 −3.2507438e−003 −2.5380632e−002 −1.5188867e−002  1.1237427e−002
−1.2165082e−002  3.6280724e−002  1.0318404e−002 −7.0397954e−003  1.2931708e−002
−3.0554539e−001  9.5069864e−001  2.5921091e−015

 1.4472794e−001  3.7194019e−002 −5.0115173e−001  3.9248514e−001 −4.3636255e−003
−1.6370609e−001 −7.3729150e−001  1.4366102e−002  3.4300789e−002 −1.3148023e−002
−1.6996325e−002  1.4513189e−002  1.4835054e−016

2.9165676e−001  1.3558307e−001  1.4263897e−001  7.8491981e−001 −3.4046475e−001
8.0371830e−002  3.6787725e−001 −4.6071001e−002 −9.4191935e−003 −4.6442990e−005
7.0991994e−003  1.1720567e−002  8.9975463e−018

 6.1420134e−002 −6.7873169e−002 −1.0654576e−002 −7.1779908e−002 −1.2683114e−001
−2.1861771e−001 −2.4820441e−003 −2.7873306e−001 −3.0264290e−001 −1.9691447e−001
8.0444279e−001  2.5938500e−001 −3.8549756e−015

−4.1124893e−001  2.5553938e−001 −2.6511808e−001  1.2780585e−001  1.7975923e−001
4.3719140e−001  5.5309088e−002 −5.7365178e−001 −2.5349274e−001  2.4176106e−001
−2.3266564e−002 −1.0887636e−002  6.1449227e−017

 3.6364609e−001  5.7663789e−001  1.8625822e−001  2.5031904e−002  6.4640829e−001
−1.8242571e−001  1.5044952e−002  7.4569698e−002 −2.0524027e−001  1.0532447e−002
2.7663294e−002  4.6875260e−003 −1.1544125e−016

−3.2339244e−024 −1.8594188e−022 −2.2864208e−020  1.8839770e−017  1.3851194e−017
6.1389926e−018  5.3793135e−018 −1.0308962e−017 −2.4777630e−017 −1.8360853e−016
−5.1809869e−015  1.0111187e−015 −1.0000000e+000

−4.1108896e−001  3.0410116e−001 −2.6475375e−002  1.1830933e−001  3.9580964e−002
1.8400086e−001 −1.6138529e−003  2.3126750e−001  1.2872830e−001 −7.8419173e−001
5.1059223e−002  2.8024804e−002 −1.2589862e−016

2.8172927e−001  3.9676858e−001 −1.1186833e−001 −2.8545475e−001 −2.3523844e−001
−6.2021941e−002  3.6391343e−002 −4.6733612e−001  6.1663861e−001 −9.7701329e−002
−2.2448637e−002 −6.9762718e−004  3.3403463e−017

−6.5036595e−002  5.4345012e−002 −6.6311998e−002  8.5899229e−002  1.8825711e−001
3.5492363e−001  1.5638708e−002  3.7547889e−001  4.9451762e−001  3.9348241e−001
5.0419384e−001  1.5752111e−001 −2.5054743e−015

−5.2505321e−001  4.1938953e−001  2.3565588e−001  5.4013256e−002 −3.0452888e−001
−5.0857011e−001 −1.0219453e−001  1.0948402e−001  1.4071860e−002  3.3981854e−001
2.0590347e−002  8.3627883e−003 −1.2167489e−016

2.3800809e−001  3.8605662e−001 −2.0893805e−001 −3.1862352e−001 −4.7293782e−001
3.5796977e−001 −5.1272835e−002  3.5840730e−001 −3.9888248e−001  8.4352298e−002
1.7215041e−003 −3.6938787e−003  8.0640562e−018
```

And the corresponding principal component variances are:

```
1.0673186e+000  8.8369309e−001  1.9208413e−001  1.3203122e−001
9.5739293e−002  6.9743565e−002  4.0237894e−002  2.6325880e−002
1.7868311e−002  3.9214603e−003  1.1638117e−003  1.0293748e−003
9.1297244e−035
```